ASG
Software Solutions

# ASG-Bridge™
# Installation Guide

Version 7.0
Publication Number: BCX0300-70
Publication Date: February 2003

ASG Worldwide Headquarters Naples Florida USA | asg.com | info@asg.com
1333 Third Avenue South, Naples, Florida 34102 USA Tel: 941.435.2200 Fax: 941.263.3692 Toll Free: 800.932.5536 (USA only)

# Contents

# Preface

This *ASG-Bridge Installation Guide* is a guide to customizing ASG-Bridge (herein called Bridge). Bridge provides solutions for Batch and CICS applications written in COBOL, PL/I and Assembler that use VSAM and sequential files. Bridge is a dynamic bridging tool that builds non-intrusive Bridge Routines that interface between converted programs and non-converted files.

Allen Systems Group, Inc. (ASG) provides professional support to resolve any questions or concerns regarding the installation or use of any ASG product. Telephone technical support is available around the world, 24 hours a day, 7 days a week.

ASG welcomes your comments, as a preferred or prospective customer, on this publication or on any ASG product.

## About this Publication

This publication consists of these chapters:

- Chapter 1, "Introduction," describes Bridge's components, purpose, and benefits.
- Chapter 2, "Customizing Bridge," describes customization of specific Bridge components, invoking Bridge, and installation validation.

# Related Publications

The documentation library for ASG-Bridge consists of these publications (where *nn* represents the product version number):

- *ASG-Bridge Installation Guide* (BCX0300-*nn*) provides information on installation and maintenance of ASG-Bridge.
- *ASG-Bridge User's Guide* (BCX0200-*nn*) contains information regarding this product, which enables field expansion for program source code, without being required to simultaneously expand the fields in files or databases.
- *ASG-Center Installation Guide* (CNX0300-*nn*) contains installation and maintenance information for ASG-Center, the common set of libraries shared by the ASG-Existing Systems Workbench suite of products.
- *ASG-ESW Enhancement Summary* (ESW1000-*nn*) highlights the new functionality for this release.

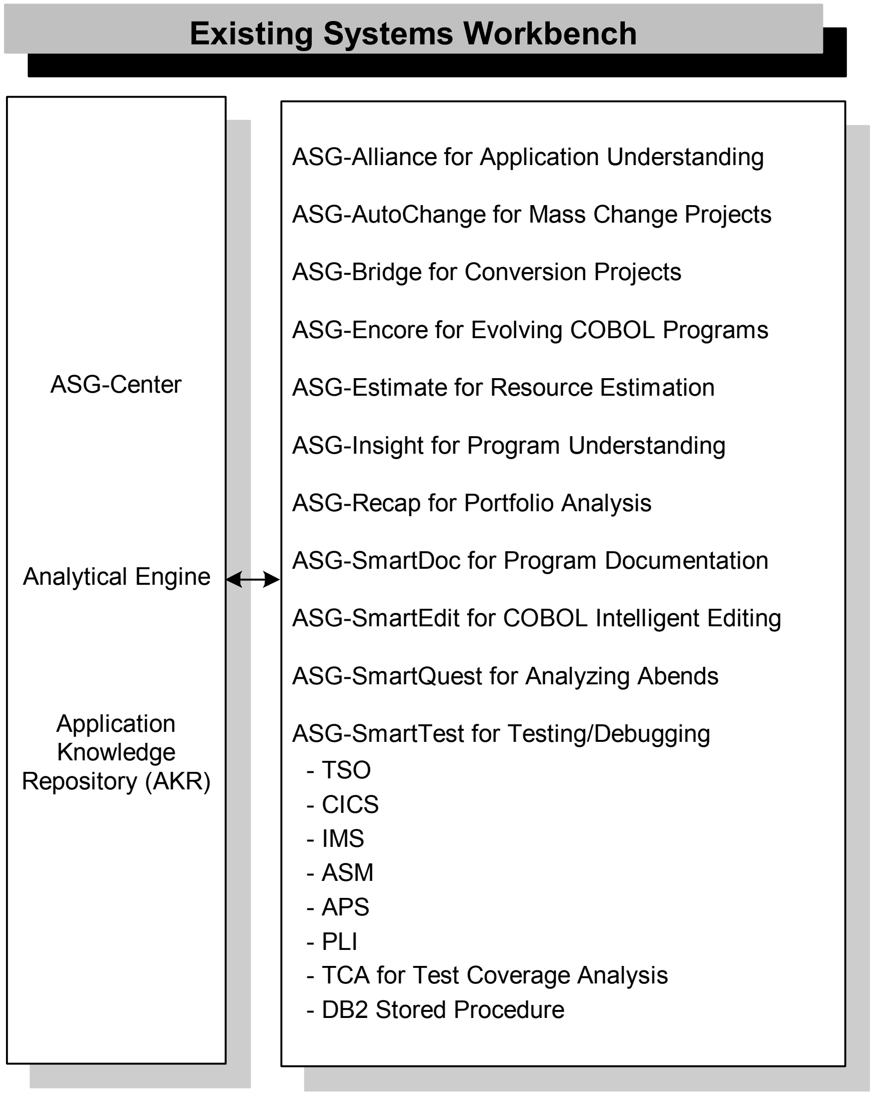
**Note:**

To obtain a specific version of a publication, contact ASG Customer Support.

# ASG-Existing Systems Workbench (ASG-ESW)

ASG-ESW (herein called ESW) is an integrated suite of components designed to assist organizations in enhancing, redeveloping, or re-engineering their existing systems. ESW products use the Application Knowledge Repository (AKR) to store source program analysis information generated by the Analytical Engine. Figure 1 represents the components of ESW.

**Figure 1 • ASG Existing Systems Workbench**


Existing Systems Workbench
ASG-Center
Analytical Engine
Application Knowledge Repository (AKR)
ASG-Alliance for Application Understanding
ASG-AutoChange for Mass Change Projects
ASG-Bridge for Conversion Projects
ASG-Encore for Evolving COBOL Programs
ASG-Estimate for Resource Estimation
ASG-Insight for Program Understanding
ASG-Recap for Portfolio Analysis
ASG-SmartDoc for Program Documentation
ASG-SmartEdit for COBOL Intelligent Editing
ASG-SmartQuest for Analyzing Abends
ASG-SmartTest for Testing/Debugging
- TSO
- CICS
- IMS
- ASM
- APS
- PLI
- TCA for Test Coverage Analysis
- DB2 Stored Procedure

This table contains the name and description of each ESW component:

| ESW Product | Herein Called | Description |
|---|---|---|
| ASG-Alliance | Alliance | The application understanding component that is used by IT professionals to conduct an analysis of every application in their environment. Alliance supports the analysis and assessment of the impact of change requests upon an entire application. Alliance allows the programmer/ analyst to accurately perform application analysis tasks in a fraction of the time it would take to perform these tasks without an automated analysis tool. The impact analysis from Alliance provides application management with additional information for use in determining the resources required for application changes. |
| ASG-AutoChange | AutoChange | The COBOL code change tool that makes conversion teams more productive by enabling quick and safe changes to be made to large quantities of code. AutoChange is an interactive tool that guides the user through the process of making source code changes. |
| ASG-Bridge | Bridge | The bridging product that enables field expansion for program source code, without being required to simultaneously expand the fields in files or databases. Because programs are converted in smaller groups, or on a one-by-one basis, and do not require file conversion, testing during the conversion process is simpler and more thorough. |
| ASG-Center | Center | The common platform for all ESW products. Center provides the common Analytical Engine to analyze the source program and store this information in the AKR. This common platform provides a homogeneous environment for all ESW products to work synergistically. |

| ESW Product | Herein Called | Description |
|---|---|---|
| ASG-Encore | Encore | The program re-engineering component for COBOL programs. Encore includes analysis facilities and allows you to extract code based on the most frequently used re-engineering criteria. The code generation facilities allow you to use the results of the extract to generate a standalone program, a callable module, a complement module, and a CICS server. Prior to code generation, you can view and modify the extracted Logic Segment using the COBOL editor. |
| ASG-Estimate | Estimate | The resource estimation tool that enables the user to define the scope, determine the impact, and estimate the cost of code conversion for COBOL, Assembler, and PL/I programs. Estimate locates selected data items across an application and determines how they are used (moves, arithmetic operations, and compares). Time and cost factors are applied to these counts, generating cost and personnel resource estimates. |
| ASG-Insight | Insight | The program understanding component for COBOL programs. Insight allows programmers to expose program structure, identify data flow, find program anomalies, and trace logic paths. It also has automated procedures to assist in debugging program abends, changing a computation, and resolving incorrect program output values. |
| ASG-Recap | Recap | The portfolio analysis component that evaluates COBOL applications. Recap reports provide function point analysis and metrics information, program quality assessments, intra-application and inter-application comparisons and summaries, and historical reporting of function point and metrics information. The portfolio analysis information can also be viewed interactively or exported to a database, spreadsheet, or graphics package. |
| ASG-SmartDoc | SmartDoc | The program documentation component for COBOL programs. SmartDoc reports contain control and data flow information, an annotated source listing, structure charts, program summary reports, exception reports for program anomalies, and software metrics. |

| ESW Product | Herein Called | Description |
|---|---|---|
| ASG-SmartEdit | SmartEdit | The COBOL editing component that can be activated automatically when the ISPF/PDF Editor is invoked. SmartEdit provides comprehensive searching, inline copybook display, and syntax checking. SmartEdit allows you to include an additional preprocessor (for example, the APS generator) during syntax checking. SmartEdit supports all versions of IBM COBOL, CICS, SQL, and CA-IDMS. |
| ASG-SmartQuest | SmartQuest | The diagnostic tool for analyzing batch and CICS transaction abends. SmartQuest has been designed to make the maximum use of simple point-and-shoot techniques to enable fast and easy navigation through any data dump. |
| ASG-SmartTest | SmartTest | The testing/debugging component for COBOL, PL/I, Assembler, and APS programs in the TSO, MVS Batch, CICS (including file services), and IMS environments. SmartTest features include program analysis commands, execution control, intelligent breakpoints, test coverage, pseudo code with COBOL source update, batch connect, disassembled object code support, and full screen memory display. |

# Invoking ESW Products

The method you use to invoke an ESW product depends on your system setup. If you need assistance to activate a product, see your systems administrator. If your site starts a product directly, use the ISPF selection or CLIST as indicated by your systems administrator. If your site uses the ESW screen to start a product, initiate the ESW screen using the ISPF selection or CLIST as indicated by your systems administrator and then typing in the product command on the command line.

The product names can also vary depending on whether you access a product directly or through ESW. See "ESW Product Integration" on page ix for more information about using ESW.

To initialize ESW products from the main ESW screen, select the appropriate option on the action bar pull-downs or type the product shortcut on the command line.

| Product Name (ESW Name) | Shortcut | ESW Pull-down Options |
|---|---|---|
| Alliance (Application Understanding) | AL | Understand ▶ Application |
| AutoChange (Conversion Set) | CC | Change ▶ Conversion Set |
| Bridge | BR | Change ▶ ASG-Bridge |
| Encore (Program Re-engineering) | EN | Re-engineer ▶ Program |
| Estimate | ES | Measure ▶ ASG-Estimate |
| Insight (Program Understanding) | IN | Understand ▶ Program |
| Recap (Portfolio Analysis) | RC | Measure ▶ Portfolio |
| SmartDoc (Program Documentation) | DC | Document ▶ Program |
| SmartEdit | SE | Change ▶ Program<br>**Or**<br>Change ▶ Program with Options |
| SmartQuest | SQV | Understand ▶ Abend/Dump |
| SmartTest (Testing/Debugging) | ST | Test ▶ Module/Transaction |

## ESW Product Integration

Because ESW is an integrated suite of products, you are able to access individual ESW products directly, or through the main ESW screen. As a result, different fields, values, action bar options, and pull-down options display on a screen or pop-up depending on how you accessed the screen or pop-up.

Certain ESW products also contain functionality that interfaces with other ESW products. Using SmartTest as an example, if Alliance is installed, SmartTest provides a dynamic link to Alliance that can be used to display program analysis information. If Insight is installed and specified during the analyze, the Insight program analysis functions are automatically available for viewing logic/data relationships and execution path. For example, the Scratchpad option is available on the Options pull-down if you have Insight installed.

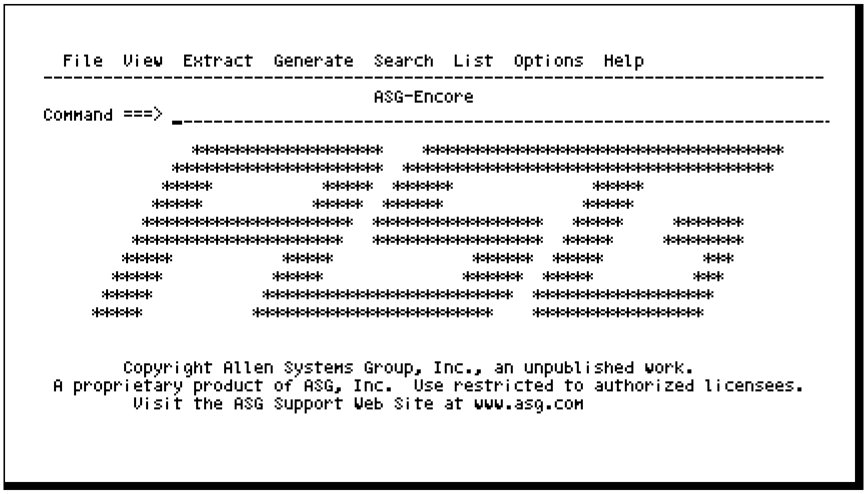
Access to these integrated products requires only that they be installed and executed in the same libraries.

## Example 1

Figure 2 shows the Encore Primary screen that displays when you access Encore directly.

The Encore Primary screen contains these eight action bar menu items: File, View, Extract, Generate, Search, List, Options, and Help.

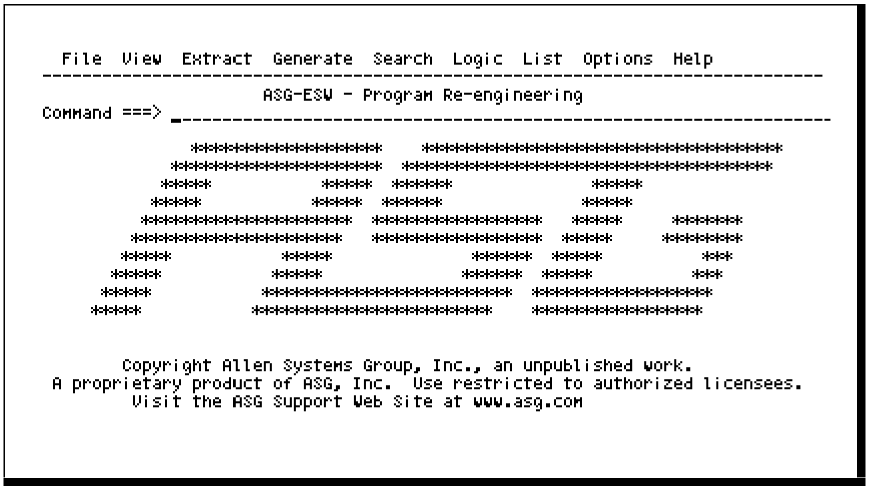
**Figure 2 • Encore Primary Screen**



```
  File  View  Extract  Generate  Search  List  Options  Help
------------------------------------------------------------------------------
                                 ASG-Encore
Command ===> _----------------------------------------------------------------

               *******************    ***********************************
             *********************  *************************************
            *****          *****  ******              *****
           *****          *****  ******              *****
          *********************  *****************   *****     *******
         *********************   *****************  *****     ********
        *****          *****              ******  *****          ***
       *****          *****              ******  *****          ***
      *****          *************************  ******************
     *****          ************************    *****************


        Copyright Allen Systems Group, Inc., an unpublished work.
 A proprietary product of ASG, Inc.  Use restricted to authorized licensees.
         Visit the ASG Support Web Site at www.asg.com
```

Figure 3 shows the Encore Primary screen that displays when you access Encore through ESW by selecting Re-engineer ▸ Program from the ESW action bar menu. Notice that the Primary screen name changes to ASG-ESW - Program Re-engineering when you enter Encore through ESW. Also, the Logic menu item displays if Insight is installed.

**Figure 3 • ESW Encore Primary Screen**

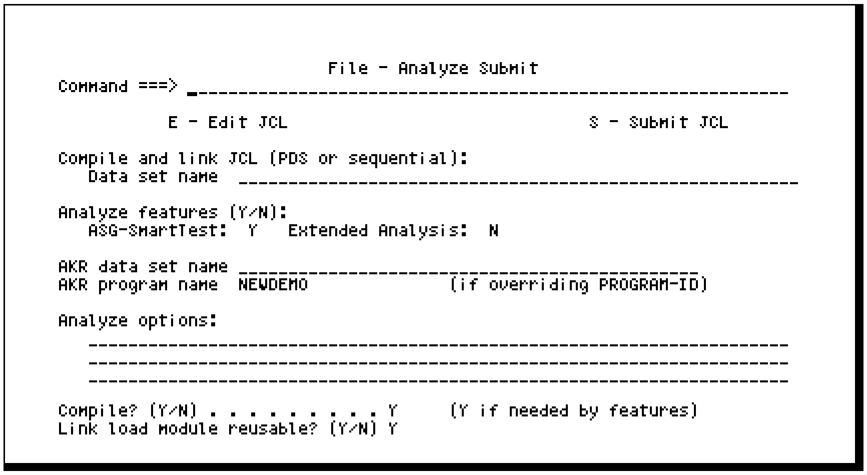


```
   File  View  Extract  Generate  Search  Logic  List  Options  Help
 ------------------------------------------------------------------------------
                        ASG-ESW - Program Re-engineering
 Command ===> _________________________________________________________________


          Copyright Allen Systems Group, Inc., an unpublished work.
  A proprietary product of ASG, Inc.  Use restricted to authorized licensees.
          Visit the ASG Support Web Site at www.asg.com
```

## Example 2

Figure 4 shows the File - Analyze Submit pop-up that displays when you access SmartTest directly. Figure 5 on page xii shows the File - Analyze Submit pop-up that displays when you access SmartTest through ESW.

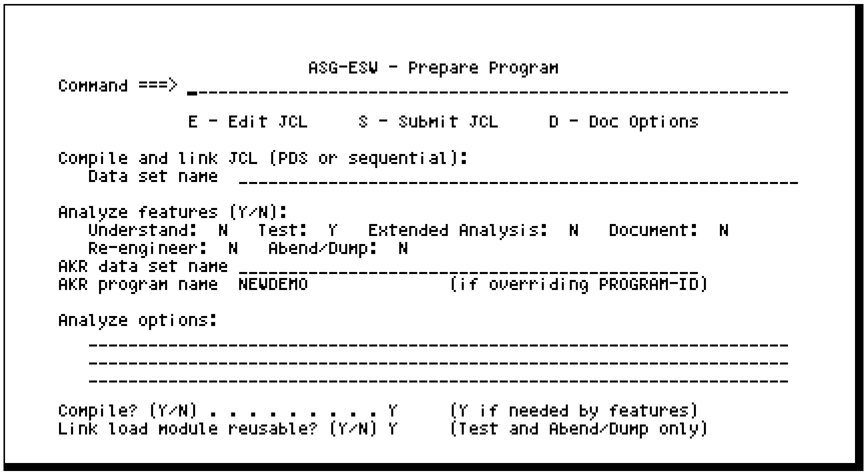
**Figure 4 • File - Analyze Submit Screen**



```
                           File - Analyze Submit
 Command ===> __________________________________________________________

           E - Edit JCL                          S - Submit JCL

 Compile and link JCL (PDS or sequential):
    Data set name  ________________________________________________________

 Analyze features (Y/N):
    ASG-SmartTest:  Y   Extended Analysis:  N

 AKR data set name ____________________________________________
 AKR program name  NEWDEMO              (if overriding PROGRAM-ID)

 Analyze options:
    ______________________________________________________________________
    ______________________________________________________________________
    ______________________________________________________________________

 Compile? (Y/N) . . . . . . . . . Y     (Y if needed by features)
 Link load module reusable? (Y/N) Y
```

The actions shown on these screens can also vary. For example, the D - Doc Options action is only available on the File Prepare Program screen (or File - Analyze Submit screen) if SmartDoc is installed on your system. In Figure 4 on page xi, the Doc Options action is not displayed.

**Figure 5 • ASG-ESW - Prepare Program Screen (accessed through ESW)**



```
                         ASG-ESW - Prepare Program
Command ===> ____________________________________________________________

             E - Edit JCL     S - Submit JCL     D - Doc Options

Compile and link JCL (PDS or sequential):
   Data set name  _________________________________________________________

Analyze features (Y/N):
   Understand:  N   Test:  Y   Extended Analysis:  N   Document:  N
   Re-engineer:  N   Abend/Dump:  N
AKR data set name ______________________________________________
AKR program name  NEWDEMO              (if overriding PROGRAM-ID)

Analyze options:
   ______________________________________________________________________
   ______________________________________________________________________
   ______________________________________________________________________

Compile? (Y/N) . . . . . . . . . Y     (Y if needed by features)
Link load module reusable? (Y/N) Y     (Test and Abend/Dump only)
```

Notice that the Analyze features field in Figure 5 lists additional ESW products than shown on Figure 4 on page xi. This field is automatically customized to contain the ESW products you have installed on your system. These are the names of the analyze types:

| Analyze Type | Analyze Type (ESW) |
|---|---|
| ASG-Encore | Re-engineer |
| ASG-Insight | Understand |
| ASG-SmartDoc | Document |
| ASG-SmartQuest | Abend/Dump |
| ASG-SmartTest | Test |
| Extended Analysis (ASG-SmartTest with Insight installed) | Extended Analysis |

# Publication Conventions

ASG uses these conventions in technical publications:

| Convention | Represents |
|---|---|
| ALL CAPITALS | Directory, path, file, dataset, member, database, program, command, and parameter names. |
| Initial Capitals on Each Word | Window, field, field group, check box, button, panel (or screen), option names, and names of keys. A plus sign (+) is inserted for key combinations (e.g., Alt+Tab). |
| *`lowercase italic monospace`* | Information that you provide according to your particular situation. For example, you would replace *`filename`* with the actual name of the file. |
| `Monospace` | Characters you must type exactly as they are shown. Code, JCL, file listings, or command/statement syntax.<br>Also used for denoting brief examples in a paragraph. |
| Vertical Separator Bar ( \| ) with underline | Options available with the default value underlined (e.g., Y\|N). |
| Underline | Denotes a cursor-selectable field or line. |

# ASG Customer Support

ASG provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. We provide all levels of support during normal business hours and emergency support during non-business hours.

**ASG Third-party Support.** ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

## Intelligent Support Portal (ISP)

Online product support is available at: http://www.asg.com/support/support.asp via the ASG Intelligent Support Portal (ISP). Your logon information for ISP online support is:

Customer ID = *NNNNNNNNN*
Password = *XXXXXXXXXX*

where:

*NNNNNNNNNN* is your customer ID supplied by ASG Product Distribution.
*XXXXXXXXXX* is your unique password supplied by ASG Product Distribution.

The *ASG-Intelligent Support Portal User's Guide* provides instructions on how to use the ISP and is located on the ASG Support web page.

## Telephone Support

To expedite response time, please have this information ready:

- Product name, version number, and release number
- List of any fixes currently applied
- Any alphanumeric error codes or messages written precisely as displayed
- A description of the specific steps that immediately preceded the problem
- Verify whether you received an ASG Service Pack or cumulative service tape for this product. It may include information to help you resolve questions regarding installation of this ASG product. The Service Pack instructions are in a text file on the distribution media included with the Service Pack. You can access the latest software corrections and Service Packs via the ISP.
- The severity code (ASG Customer Support uses an escalated severity system to prioritize service to our clients. The severity codes and their meanings are listed below.)

**Severity Codes and Expected Support Response Times**

| Severity | Meaning | Expected Support Response Time |
|---|---|---|
| 1 | Production down, critical situation | Within 30 minutes |
| 2 | Major component of product disabled | Within 2 hours |
| 3 | Problem with the product, but customer has work-around solution | Within 4 hours |
| 4 | "How-to" questions and enhancement requests | Within 4 hours |

## The Americas

| | Phone | Fax | E-mail |
|---|---|---|---|
| **United States and Canada** | 800.354.3578 | 1.703.464.4901 | support@asg.com |

## Europe, Middle East, and Africa (EMEA)

During normal business hours, we recommend that you call the Central Support number first (except in South Africa).

| | Phone | Fax | E-mail |
|---|---|---|---|
| **Central Support** | 00.800.3544.3578 | 44.1727.812018 | support.emea@asg.com |
| **English** | 44.1727.736305 | 44.1727.812018 | support.uk@asg.com |
| **French** | 33.141.028590 | 33.141.028589 | support.fr@asg.com |
| **German** | 49.89.45716.200 | 49.89.45716.400 | support.de@asg.com |
| **Italian** | 39.0290450025 | | support.it@asg.com |
| **Dutch** | 31.30.241.6133 | | support.nl@asg.com |
| **Spanish** | 34.913.523.800 | 34.917.156.961 | support.es@asg.com |
| **South Africa** | 800.201.423 | | support.sa@asg.com |

## Asia Pacific (APAC)

| | Phone | Fax | E-mail |
|---|---|---|---|
| **Central Support** | 61.3.9645.8500 | 61.3.9645.8077 | support.au@asg.com |
| **Australia** | 800.637.947 | 61.3.9645.8077 | support.au@asg.com |
| **Hong Kong** | 800.96.2800 | | support.hk@asg.com |
| **Japan** | 81.3.5326.3684 | 81.3.5326.3001 | support.au@asg.com |
| **Singapore** | 65.224.3080 | 65.224.8516 | support.sg@asg.com |

## All Other Countries (Also for any non-working numbers)

| | Phone | Fax | E-mail |
|---|---|---|---|
| **All other countries** | 1.239.435.2201 | | support@asg.com |

If you receive a voice mail message, follow the instructions to report a production-down or critical problem. Leave a detailed message including your name and phone number. An ASG Customer Support representative will be paged and will return your call as soon as possible. Please have available the information described previously when the ASG Customer Support representative contacts you.

# ASG Documentation/Product Enhancements

Submit all product and documentation suggestions to ASG's product management team at http://www.asg.com/asp/emailproductsuggestions.asp.

If you do not have access to the web, FAX your suggestions to product management at (239) 263-3692. Please include your name, company, work phone, e-mail ID, and the name of the ASG product you are using. For documentation suggestions include the publication number located on the publication's front cover.

# 1 Introduction

This chapter introduces Bridge and contains these sections:

## ASG Service Pack

Verify whether you received an ASG Service Pack for this product. If so, read the instructions for installing the Service Pack before proceeding with the product installation. The installation instructions are located in a text file on the distribution media included with the Service Pack. If you have any problems with the Service Pack, contact ASG Customer Support.

## What is Bridge?

Bridge is a dynamic bridging tool that builds non-intrusive Bridge Routines that communicate between converted programs and non-converted files. Bridge automatically invokes Bridge Routines to convert records read and written from non-converted files during program execution, deferring the need for you to convert files. Bridge consists of an online, interactive administration facility, and a dynamically invoked automatic bridging component.

Bridging is a compliance conversion strategy that is program-centric and facilitates individual or group program conversions without requiring you to convert files. Converted programs process unconverted files by converting the file records while they read and write them. The file remains in an unconverted state.

## Benefits of Bridge

These are the benefits of using Bridge for your conversion project:

- Simplifies conversion planning. Planning becomes program-centric instead of data-centric.
- Minimizes the complexity and risk of conversions.
- Defers data file conversion deadlines indefinitely.
- Reduces the coordination problems of using EDI (electronic data interchange) and other outside data.
- Allows compliant and non-compliant programs to share the same data before and after conversion.
- Guarantees seamless access to non-compliant historical and backup data.

## Implementing Bridge

Bridge requires you to define the structure of each file and its respective conversion fields. You can then use these definitions to generate executable bridging routines to convert records the program reads and writes. These executable Bridge Routines are external to the application programs.

Bridge automatically intercepts all program I/O operations and converts data to and from the new format as needed. The program gets a converted view of the non-converted file.

These are the phases to implement Bridge technology:

- Generate the Bridge Routines from user-defined Bridge Definitions.
- Integrate the Bridge Routines with a converted program.

# 2 Customizing Bridge

This chapter describes customizing your Bridge installation and contains these sections:

## Prerequisite

You must install and customize Center before customizing Bridge. If Center is not installed, see the *ASG-Center Installation Guide*.

# Using Bridge Parameter and Rule Starter Sets

You define and maintain Bridge Parameters and Rules. However, Bridge provides a starter set of parameters and rules you can use to expedite and simplify your first use of the product.

Use the import function to import the starter sets from these members in the ESW CNTL library after Bridge is installed:

**Note:**

Always import the parameter file before you import the corresponding rule file.

| Parameter | Control Library Member |
|---|---|
| Basic Bridge Parameters | VIAGBPSS |
| Basic Bridge Rules | VIAGBRSS |
| Bridge 2000 Parameters | VIAGYPSS |
| Bridge 2000 Rules | VIAGYRSS |
| Bridge for Euro Parameters | VIAGEPSS |
| Bridge for Euro Rules | VIAGERSS |

For information on how to use the import function, see Importing Bridge Parameters, Rules, or Definitions in the *ASG-Bridge User's Guide*.

# Installing and Validating Bridge

## Installation Checklist

Company: ______________________________

Installer: ______________________________

Date: ________ / ________ / ________

Complete these steps for Bridge:

| OK | Control Members | Step/Page | Opt/Req |
|---|---|---|---|
|  | Specify the installation options.<br>Edit VIA$PRMG and VIA$PRMU to override the installation defaults. | 1a / 11 | Required |
|  | Specify the run time parameters.<br>Edit VIAGPARM, then run the job VIAGASMP to generate the module VIAGBRGP. | 1b / 12 | Required |
|  | Install the Extended SVC.<br>Edit and submit VIAGLPAC to copy IGX00SVC as IGX00204. | 1c / 18 | Required |
|  | Define the Bridge management in the AKR.<br>Edit and submit VIAGAKRJ to allocate the base AKR. | 1d / 19 | Required |
|  | Define the Bridge Event Log.<br>Edit and submit VIAGABLJ to allocate the Bridge Event Log dataset. | 1e / 19 | Optional |
|  | Customize Panvalet, Librarian, or User Source Manager.<br>Edit and submit VIASPAMJ to customize the Panvalet source manager.<br>Edit and submit VIASFAIJ or VIASFAIR to customize the Librarian source manager. | 1f / 20 | Optional |

| OK | Control Members | Step/Page | Opt/Req |
|---|---|---|---|
| | Add Bridge Modules to MLPA/PLPA.<br>Submit jobs VIASLPAJ and VIASLPXJ to move VIAGMAIN to MLPA/LPA. | 1g / 21 | Optional |
| | Update the Message Route Code.<br>Edit VIAGMSG, then edit and submit VIAGMSGA to alter Route Code. | 1h / 22 | Optional |
| | Generate the currency table load module.<br>Run VIAGECUT to generate the currency table load module (for Bridge for Euro option only). | 1i / 22 | Optional |
| | Prepare for Batch verification.<br>Run job VIAGAIVB to allocate the files for the sample programs. | 3a / 24 | Required |
| | Verify the conversion of the sequential files.<br>Run the job VIAGICBJ to verify the conversion of the sequential files. | 3b / 25 | Required |
| | Verify the conversion of the key sequenced (KSDS) VSAM files processed through the sequential access mode.<br>Run VIAGIVSJ to verify the conversion of the KSDS VSAM files that are using sequential access. | 3c / 25 | Required |
| | Verify the conversion of the key sequenced (KSDS) VSAM files accessed through random access mode.<br>Run VIAGIVRJ to verify the conversion of the KSDS VSAM files that are using random access. | 3d / 26 | Required |
| | Verify the conversion of currency table (Bridge for Euro Batch option only).<br>Run VIAGEBIJ for euro Batch and VIAGEBVJ for euro VSAM. | 3e / 26 | Optional |

If you installed the CICS Support option or the CICS Support option with DL/I, complete these steps appropriate to the CICS option you are installing:

| OK | Control Members | Step/Page | CICS or CICI DL/I |
|---|---|---|---|
| | Define the CICS definitions.<br>Run job VIAGCSDI to allocate the programs, transactions, and files.<br>Run job VIAGCSDD to allocate the programs, transactions, and files. | 1a / 27 | Both<br>CICS DL/I |
| | Install the Bridge group.<br>Add Group (VIAGCICS) to your GRPLIST. If you do not perform this step, you must use CEDA to install this group after you restart CICS. | 1b / 28 | Both |
| | Add group (VIAGDLI) to your GRPLIST. If you do not perform this step, you must use CEDA to install this group after you restart CICS. | 1b / 28 | CICS DL/I |
| | Activate Bridge at CICS startup.<br>Add PLT entries (VIAGPLTI) to PLTPI.<br>**Or**<br>Add B2KD/ON\ to the SEQ INPUT device. | 2 / 28 | Both |
| | Add PLT entries (VIAGPLTI) to PLTPI.<br>**Or**<br>Add B2KD/ON\ to the SEQ INPUT device. | 2 / 28 | CICS DL/I |
| | Allocate the installation verification file.<br>Run job VIAGAIVC to allocate files for the IVP. | 3a / 30 | CICS |
| | Allocate and populate the IVP libraries.<br>Run job VIAGICPD to allocate and populate the IVP libraries. | 3b / 30 | CICS DL/I |

| OK | Control Members | Step/Page | CICS or CICI DL/I |
|---|---|---|---|
| | Allocate and initialize the sample IVP database.<br>Run jobs VIAGIDFJ and VIAGIDRJ to allocate and initialize the sample IVP database for forward and reverse bridging. | 3c / 31 | CICS DL/I |
| | Generate the Bridge ACBs.<br>Run job VIAGACBJ to generate the ACBs for VIAGDIVF and VIAGDIVR. | 3d / 31 | CICS DL/I |
| | Start the Trace Facility. | 3e / 31 | CICS DL/I |
| | Edit the CICS startup JCL. | 4 / 32 | Both |
| | Add Bridge entries to CICS DL/I database directory (DDIR).<br>Add the DSNs for DHFACULR, DPFACULR, and DPFACULT to CICS JCL. | 5 / 32 | CICS DL/I |
| | Add the Bridge entries to CICS DL/I PSB Directory (PDIR). | 6 / 32 | CICS DL/I |
| | Restart CICS. | 7 / 33 | Both |
| | Verify the CICS installation.<br>Run transactions B2KI (forward bridging) and B2KR (reverse bridging). | 8 / 33 | CICS |
| | Run the IVP transactions.<br>Run transactions B2KC (forward bridging) and B2KV (reverse bridging). | 9 / 33 | CICS DL/I |
| | Run the IVP transactions for CICS RLAPI.<br>Run transaction BRIV. | 9a / 34 | CICS DL/I |
| | Run the IVP transactions for Bridge Euro.<br>Run transaction BEUR. | 9b / 34 | CICS DL/I |

If you installed the IMS Support option, complete these tasks:

| OK | Control Members | Step/Page |
|---|---|---|
| | Allocate and populate the IMS IVP libraries.<br>Run job VIAGICPY to allocate and populate the IMS IVP libraries. | 1 / 35 |
| | Allocate and initialize the IMS IVP database.<br>Run job VIAGILIJ to allocate and initialize the IMS IVP databases. | 2 / 35 |
| | Run the IMS IVP program.<br>Run job VIAGIRIJ to run the IMS installation verification program. | 3 / 35 |
| | Run the IMS/HSSR IVP program.<br>Run job VIAGIHRJ to run the IMS/HSSR installation verification program. | 4 / 36 |

If you installed the IMS FastPath Support option complete these tasks:

| OK | Control Members | Step/Page |
|---|---|---|
| | Define the transaction, database, and PSB.<br>Definitions are contained in CNTL library member VIAGXTRN. | 1 / 36 |
| | Perform a DBDGEN, PSBGEN, and ACBGEN for the supplied DBD and PSB.<br>CNTL library members VIAGXFPD and VIAGXFPP contain the source for the DBD and PSB. | 2 / 37 |
| | Perform MFSGEN to install the supplied MFS format.<br>CNTL library member VIAGXMFS contains the source of the MFS format. | 3 / 37 |
| | Create the MSDB database initialization file.<br>CNTL library member VIAGXDBJ contains the sample JCL. | 4 / 37 |
| | Create the JCL for executing IFP region for IVP transaction program.<br>CNTL library member VIAGXFPJ contains the sample JCL. | 5 / 38 |

| OK | Control Members | Step/Page |
|---|---|---|
| | Perform Bridge DCRGEN and PCRGEN for the supplied DBD and PSB.<br>Run DCRGEN by using CNTL library member VIAGXSAL as the copybook and VIAGXFDC as the DCD.<br>Run DBDGEN and PSBGEN while referencing the CNTL library members VIAGXFND and VIAGXFNP. | 6 / 38 |
| | Initialize the IMS control region and invoke the IVP transaction. | 7 / 38 |

If you installed the IDMS Support option, complete these tasks:

| OK | Control Members | Step/Page |
|---|---|---|
| | Add the attributes and protocols required by Bridge to the IDMS data dictionary.<br>Load the Bridge IDMS protocol (VIAGMPRB) for Batch processing. | 1 / 39 |
| | Relink Bridge with the IDMS interface. | 2 / 40 |
| | Allocate the IDMS IVP database.<br>Run job VIAGMCPY to allocate and populate the IDMS IVP libraries. | 3a / 41 |
| | Define the IDMS database to the IDMS dictionary. | 3b / 41 |
| | Initialize and load the IDMS IVP database.<br>Run job VIAGMLIJ to allocate and initialize the IDMS IVP databases. | 3c / 42 |
| | Run the IDMS IVP program.<br>Run job VIAGMRIJ to run the IDMS installation verification program. | 3d / 43 |
| | Compile and run the IDMS-DC IVP program. | 3e / 45 |

The installation is now complete.

## Step 1 - Modifying and Executing CNTL Library Members

**Note:**

If you are installing Bridge without Center, you must use the Bridge password to download the CNTL library members.

### Step 1a - Specifying Installation Options

**Note:**

Perform this step only if you want to override default Bridge installation options.

To override the default Bridge installation options, follow this step:

- Edit installation option members VIA$PRMG and VIA$PRMU. Modify or add the appropriate option.

#### VIA$PRMG

This is a Bridge installation option parameter specified in VIA$PRMG:

```
Bridge-Generate-Max-Per-JOB=10
```

Specifies the maximum number of Bridge Definitions allowed per Bridge Definition Generate job. The default is 10.

#### VIA$PRMU

These are the Bridge installation option parameters specified in VIA$PRMU:

```
Bridge-Base-AKR-DSN='ASG.VIACENxx.BRIDGE.AKR'
```

Specifies the DSN of the Bridge Management AKR. This AKR is used to maintain information, such as the names of the generate libraries and Bridge Definition relationships. AKR information is stored in the AKR member $$BRIDGE.

```
Bridge-Log-DSN=ASG.VIACENxx.BRIDGE.LOG
```

Specifies the name of the dataset for the Bridge Event Log. The Event Log is a dataset that records significant events such as creating, renaming, deleting, importing, exporting, and generating Bridge Definitions.

If you do not want to use the Bridge Event Log, follow this step:

- Comment out this parameter (with an asterisk in the first position).

  **Or**

  Space out the dataset name to deactivate it.

To activate the Bridge Event Log, follow this step:

- Specify the parameter with a dataset name and make sure the dataset is allocated and initialized. The next time you start Bridge the Bridge Event Log is activated.

If you specify a Bridge Event Log dataset name, but Bridge cannot find it, a message displays when the product starts stating that the Bridge Event Log could not be found and that event logging will not be performed.

The Bridge Event Log record is a comma-delimited file containing the date, time, user ID, AKR name, Bridge name, and action. JCL is provided to allocate and initialize the Bridge Event Log in a subsequent step of this installation process. You can browse the Bridge Event Log, or input it into a program that processes comma-delimited files to produce an activity report.

## Step 1b - Specifying Run-time Parameters

In this step you create the Bridge run-time parameters module. Modify member VIAGPARM as necessary for your installation and assemble to generate the run-time parameters module. VIAGPARM contains installation-defined parameters that Bridge uses during execution. These are some of the parameters:

- CICSLOCALOPEN
- SVC 109 Extended Services Routine function code to be used by Bridge
- Bridge User Exit names
- The DD names and dataset names of libraries used by the Bridge IMS support

### *To specify the parameters*

1. Edit the VIAGPARM CNTL member, supplying the indicated parameters.
2. Complete and save changes.
3. Edit the member VIAGASMP.
4. Verify the JOB statement and submit the job to compile the module.
5. Verify that the job ends with a completion code of 0.

These parameters are Assembler macro statement operands and must conform to these syntax rules of Assembler language:

- Each parameter must be followed by a comma if there are additional parameters to follow.
- Multiple parameters can appear in a single statement as long as you separate them with single commas.
- Each statement must begin in column 16, and if another statement follows, the last parameter in the statement must be followed with a comma.
- Column 72 must contain a non-blank character.

For your convenience, each parameter in VIAGPARM is coded on a separate statement that you can leave as is, modify, or remove.

## CICSLOCALOPEN

When Bridge is called in a CICS environment and a requested local file is not open, Bridge opens it. The CICSLOCALOPEN parameter enables you to specify whether Bridge opens a local file to verify LRECL and RECFM. This is the syntax for the CICSLOCALOPEN parameter:

```
CICSLOCALOPEN=[Y|N]
```

If you specify CICSLOCALOPEN=N, Bridge does not open the local file and the values from the DCR are used without verification. This process is similar to that used for REMOTE files.

## SVC 109 Extended Services Routine (ESR) Function Code

This parameter specifies the value of the SVC 109 function code that is to be used by Bridge:

```
SVC109=204
```

The parameter defaults to 204. If this value is already in use in your installation, specify a value between 200 and 255 that is not used in your installation.

## Bridge User Exits

**Note:**

Using Bridge user exits is optional. Skip this parameter if you decide not to install user exits.

Bridge provides user exits that allow installation-written code to perform additional functions. The use of these exits is optional. If you use one or more user exits, you must specify the name of each user exit and the name of the program the user exit invokes. This table lists the available user exits:

| Exit Name | Description |
|---|---|
| EXBTIN | Batch VSAM/sequential initialization complete |
| EXBTOP | Batch VSAM/sequential open intercept complete |
| EXDCR | Successful Bridge Routine generation complete |
| EXCSIN | CICS environment initialization complete |
| EXIM01 | IMS open intercept complete |
| EXMP01 | IDMS environment initialization complete |

For example:

```
EXBTIN=MYEXITID,
EXBTOP=,
EXDCR=,
EXCSIN=,
EXIM01=,
```

This example specifies that the Batch Interface Initialization Complete user exit is a module named MYEXITID. No other user exits are used.

## DCR and PCR Libraries

The IMS and DL/I Support options of Bridge construct and use two types of executable Bridge Routines: DBD Conversion Routines (DCRs) and PSB Conversion Routines (PCRs). A DCR contains bridging routines and information about the DBD that it converts. A PCR contains bridging routines and information about the PSB that it converts.

DCRs have the same name as the DBDs that they convert, and PCRs have the same name as the PSBs that they convert. Consequently, DCRs and PCRs must be treated differently than DBDs and PSBs to prevent conflict when trying to load one or the other from their respective libraries. Bridge uses special DCR libraries and PCR libraries.

## Specifying the IMS and DL/I Support DCR and PCR Libraries

For IMS support, Bridge loads DCRs and PCRs from their respective libraries rather than from the normal program libraries used in your JCL. Bridge provides parameters to specify the DD names and dataset names of these libraries. For CICS DL/I support, the CICS program loader loads DCRs and PCRs into memory. The DCR and PCR libraries become part of the DFHRPL concatenation.

To avoid conflict caused by DCRs or PCRs having the same name as existing CICS programs and PSBs, Bridge supports alias names for DCRs and PCRs. If you assign an alias to a DCR or PCR, it takes precedence over its original name when Bridge loads it into memory.

Bridge permits you to use a standard or global DCR and/or PCR library where you can keep all of your production DCRs or PCRs. An installation can have only one global DCR or PCR library. If you use a global DCR library, you must specify the dataset name of this library in the DCRLIB or PCRLIB parameter.

Bridge also permits you to use one or more local DCR or PCR libraries. These libraries might contain DCRs or PCRs segregated by type (production and test), organized by application, or based upon your organization. If you decide to use local DCR or PCR libraries, you must specify the DD name that Bridge looks for in the DCRLIBDD or PCRLIBDEF parameter. You have the option of using a global DCR or PCR library, one or more local DCR or PCR libraries, or a combination of both.

When Bridge initializes its IMS environment, it attempts to allocate a local DCR or PCR library defined in the job step's JCL. If that fails, Bridge attempts to dynamically allocate the global DCR or PCR library. If that fails, the IMS environment does not initialize.

**Note:**

If you are installing the Bridge IMS Support option, one or more of these parameters may be required.

These are the DCR and PCR parameters:

| Parameter | Description |
|---|---|
| DCRLIBDD | Specifies the DD name that Bridge looks for to allocate the DCR library. If you omit this parameter, Bridge uses a default value of VIAGDCRL. |
| DCRLIB | Specifies the dataset name of global DCR library. If you omit this parameter, Bridge does not attempt to allocate a global DCR library. |

| Parameter | Description |
|---|---|
| PCRLIBDD | Specifies the DD name that Bridge looks for to allocate the PCR library. If you omit this parameter, Bridge uses a default value of VIAGPCRL. |
| PCRLIB | Specifies the dataset name of global DCR library. If you omit this parameter, Bridge does not attempt to allocate a global DCR library. |

If you specified these two parameters, Bridge attempts to open a library with the DD name of DCRLIB, for example:

```
DCRLIB DD DSN=MY.DCR.LIBRARY,DISP=SHR
```

If Bridge does not find or is unable to open this DD statement, it dynamically allocates the global DCR library of MAIN.DCR.LIBRARY, for example:

```
DCRLIBDD=DCRLIB and DCRLIB=MAIN.DCR.LIBRARY
```

You can define the DD names of local DCR and PCR libraries, and the dataset names of global DCR and PCR libraries, for example:

```
DCRLIBDD=DCRLIB,
DCRLIB=GLOBAL.DCR.LIB,
PCRLIBDD=PCRLIB,
PCRLIB=GLOBAL.PCR.LIB,
```

You can define the DD names of local DCR and PCR libraries. Global DCR and PCR libraries are not used, for example:

```
DCRLIBDD=DCRLIB,
PCRLIBDD=PCRLIB,
```

## DCR and SCR Libraries for IDMS Support

The IDMS Support option of Bridge constructs and uses two types of executable Bridge Routines: DCRs (Schema Data Conversion Routines) and SCRs (Subschema Conversion Routines). A DCR contains bridging routines and information about the IDMS schema and records that it converts. A SCR contains bridging routines and information about the subschema.Treat these Bridge Routines differently from normal programs to prevent conflict when loading one. Bridge uses special libraries for DCRs and SCRs called DCR libraries and SCR libraries, respectively.

### Specifying the IDMS Support DCR and SCR Libraries

For IDMS support, Bridge loads DCRs and SCRs from their respective libraries rather than from the normal program libraries used in your JCL. Bridge provides parameters to specify the DD names and dataset names of these libraries.

First, Bridge permits the use of a standard or global DCR and/or SCR library in which you might keep all of your production DCRs or SCRs. An installation can have only one global DCR or SCR library. If you use a global DCR or SCR library, you must specify the dataset name of this library in the DCRLIB or SCRLIB parameter.

Bridge also permits the use of one or more local DCR or SCR libraries. These libraries might contain DCRs or SCRs segregated by type (production or test), or they may be organized by application or based upon your organization. If you decide to use local DCR or SCR libraries, you must specify the DD name that Bridge looks for in the DCRLIBDD or SCRLIBDD parameter. You have the option of using a global DCR or SCR library, one or more local DCR or SCR libraries, or a combination of both.

When Bridge initializes its IDMS environment, it attempts to allocate a local DCR or SCR library defined in the job step's JCL. If it fails, Bridge then attempts to dynamically allocate the global DCR or SCR library. If that fails, the IDMS environment does not initialize.

**Note:**

If you are installing the Bridge IDMS Support option, one or more of the SCR and DCR parameters may be required.

These are the SCR and DCR parameters:

| Parameter | Description |
|---|---|
| DCRLIBDD | Specifies the DD name that Bridge looks for to allocate the local DCR library. If you omit this parameter, Bridge uses a default value of VIAGDCRL. |
| DCRLIB | Specifies the dataset name of the Global DCR library. If you omit this parameter, Bridge does not attempt to allocate the Global DCR library. |
| SCRLIBDD | Specifies the DD name that Bridge looks for to allocate the SCR library. If you omit this parameter, Bridge uses a default value of VIAGSCRL. |
| SCRLIB | Specifies the dataset name of the Global SCR library. If you omit this parameter, Bridge does not attempt to allocate the Global SCR library. |

If you specify these parameters, Bridge first attempts to open a library with the DD name of DCRLIB. Whether this attempt is successful or not, Bridge then attempts to dynamically allocate the global DCR library of GLOBAL.DCR.LIBRARY, for example:

```
DCRLIBDD=DCRLIB and DCRLIB=GLOBAL.DCR.LIBRARY
```

You can define the DD names of local DCR and SCR libraries, and the dataset names of global DCR and SCR libraries, for example:

```
DCRLIBDD=DCRLIB,
DCRLIB=GLOBAL.DCR.LIB,
SCRLIBDD=SCRLIB,
SCRLIB=GLOBAL.SCR.LIBRARY,
```

You can define the DD names of local DCR and SCR libraries. Global DCR and SCR libraries are not used, for example:

```
DCRLIBDD=DCRLIB,
SCRLIBDD=SCRLIB,
```

## Step 1c - Installing the Extended SVC

**Note:**

This step is required.

The VIAGLPAC member copies the SVC 109 extended service module into the LPA library. The data must be copied into the LPA library.

**Note:**

The name assigned to the extended service module depends on the function code number assigned in "Step 1b - Specifying Run-time Parameters" on page 12 (e.g., the function code becomes the last 3 characters of the module name). For example, if you specified a function code of 240, the name assigned to the module in this step would be IGX00240.

To update the parameters LPALIB, VIASOFT, and CENTER, follow this step:

- Edit VIAGLPAC.

### *To copy the SVC to LPA*

1 Supply a valid JOB statement and submit the job.

2 Verify that the job ends with completion code 0.

To make the extended SVC routine (ESR) available, follow this step:

- Use the CLPA initialization parameter to IPL your MVS system.

  Alternatively, you can use ASG-TMON for MVS to load the SVC routine into the LPA and define it as ESR SVC dynamically.

## Step 1d - Defining Management AKR

Bridge uses a special base AKR called the Bridge Management AKR to contain internal and installation-dependent information.

### *To create the AKR*

1. Edit the CNTL member VIAGAKRJ to specify the name and attributes of the base AKR.

   **Note:**

   This name of the base AKR is given in "Step 1a - Specifying Installation Options" on page 11 for the Bridge-Base-AKR-DSN.

2. Supply a valid JOB statement and submit the job.

   To allocate additional AKRs, you can modify and use this JCL or you can use the AKR Utility from within the Bridge interactive facility.

## Step 1e - Defining the Bridge Event Log

**Note:**

If you do not want to use the Bridge Event Log, omit the Bridge-Log-DSN parameter specified in "Step 1a - Defining CICS Option Definitions" on page 27 of this procedure and skip this step.

Bridge records significant events in an optional Bridge Event Log. The VIAGABLJ member allocates and initializes a dataset for use as a Bridge Event Log.

### *To define the Bridge Event Log*

1. Edit member VIAGABLJ to designate these items:

   - Dataset name (specified in the Bridge-Log_DSN parameter in "Step 1a - Defining CICS Option Definitions" on page 27)
   - Unit name
   - Volume ID of the Bridge Event Log to be allocated.

2. Submit the job. Verify that the job ends with a completion code 0.

The space allocation for the log should be sufficient for logging a reasonable amount of time after initialization. However, like any dataset of this type, it eventually becomes full. ASG recommends that you establish a procedure to periodically review the available space remaining and switch to an alternate, dump the log, and reinitialize it. The space allocation parameters may require adjustment to comply with your review procedure.

## Step 1f - Customizing Panvalet or Librarian Source Manager

Bridge uses your source code to map the files being bridged.

**Note:**

If your source code is stored in a PDS, you can skip this step.

### Panvalet Customization

#### *To customize Panvalet*

1. Edit the VIASDAMJ CNTL member. Specify a valid JOB card and the correct values for VIASOFT, ASMBLR, CENTER, PANLIB, SYSOUT, and SYSDA parameters.

2. Submit this job to assemble and link edit the Panvalet VIASPAM module.

**Note:**

Do not perform this step if you are using Panvalet R12 or above and the module PAM is available in LINKLST or LPA.

**Customized Edit Panel.** If your site has modified Panvalet's customized edit panel, you might need to include these changes in the ESW panels VSPENPAN, VSPE2PAN, and VSPEDPAN. If this situation occurs, contact ASG Customer Support.

**WTP Messages.** If the Panvalet load library is allocated under ISPLLIB and the LIBDEF facility is used, `MEMBER NOT FOUND` warning messages display each time you select a Panvalet member. These are the alternatives:

- Add Bridge allocations to LOGON and bypass LIBDEF processing.
- Add Panvalet load library allocation to VIALLIB in LIBDEF processing.
- Set WTPMSG OFF (not recommended).

## Librarian Customization

### *To customize Librarian*

**1** Edit the VIASFAIJ CNTL member, specify a valid JOB card and the correct values for VIASOFT, ASMBLR, CENTER, LIBRLIB, LIBRMAC, SYSOUT, and SYSDA parameters.

If you are running Librarian R3.9, edit CNTL member VIASFAIR and comment out the four lines containing these delete statements:

```
DELETE EPLOC=XFAIROPN
DELETE EPLOC=XFAIRMOD
DELETE EPLOC=XFAIRREC
DELETE EPLOC=XFAIRCLS
```

**2** Submit this job to assemble and link edit the Librarian VIASFAIR module.

**Customized Edit Panel.** If your site has modified Librarian's customized edit panel, you might need to include these changes in the ESW panels VSPENLIB, VSPEDLIB, and VSPE2LIB. If this situation occurs, contact ASG Customer Support for instructions.

**WTP Messages.** If the Librarian load library is allocated under ISPLLIB and the LIBDEF facility is used, `MEMBER NOT FOUND` warning messages display each time you select a Librarian member. These are possible alternatives:

- Add Bridge allocations to LOGON and bypass LIBDEF processing.
- Add Librarian load library allocation to VIALLIB in LIBDEF processing.
- Set WTPMSG OFF (not recommended).

## *Step 1g - Adding Bridge Modules to MLPA/PLPA*

The main load module of the interactive facility, VIAGMAIN, is re-entrant and available for location in Extended LPA. These are the advantages of using Extended LPA:

- A reduction of the memory requirement per user
- An overall decrease in required swap space
- Performance improvements

Moving this module to MLPA/PLPA is optional. ASG recommends you keep the original ESW load library (from the installation tape) as a staging library so that any required PTFs can easily be applied. You can copy the re-entrant module to Extended LPA and the non-re-entrant module to a separate user library. Also, these steps require changes to user logons or product allocations. The CNTL library contains two members, VIASLPAJ and VIASLPXJ, that you can use to perform these copy steps.

**Note:**

Do not use the ISPF 3.3 copy feature to copy these modules because some of them have aliases.

## Step 1h - Updating the Message Route Code

By default, Bridge uses route code 11, SYSMSG to send messages.

### *To change the route code*

1. Enter your value next to the ROUTCDE parameter in the VIAGMSG macro in the CNTL library.

2. If you change the message text or the route code, edit the CNTL member VIAGMSGA and supply the correct values for VIASOFT and CENTER.

3. Supply a valid JOB statement and submit the member to assemble the new message table and specify the routing code.

   **Note:**

   This step is not required unless you want to change the route code.

## Step 1i - Generating the Currency Table Load Module - Bridge for Euro Option Only

If you are using the Bridge for Euro option, run the VIAGECUT CNTL member to generate the currency table load module.

Bridge for Euro rules use the EuroConverter currency table stored in a load module, VIAGSECT. This load module is generated by running the VIAGECUT CNTL member. VIAGECUT converts a text file to Assembler statements and assembles them. A REXX EXEC, VIAGECTB, is provided in the CLIST library for implementing the currency table conversion.

You can optimize VIAGSECT by listing the most common currencies first (for example your local national currency and the euro).

**Note:**

A starter EuroConverter currency table is available in VIAGECCT.

VIAGECUT must be customized to your company’s naming standards. The DD name of the input file is ECINPUT.

The format of your input file is based on the format of the EuroConverter Studio currency table. This file is sorted by alias name. However, you can edit this file using an editor such as ISPF Edit before you run the REXX EXEC.

**Note:**

The Bridge rule passes the currency by value or by input field. If Bridge passes both, the value parameter is used.

Bridge delimits EuroConverter Studio currency table records by the character in the first position of the file. In this example, a semicolon is used. This is the input file syntax:

```
    ;EUR  ;EUR   ;Euro          ;1      ;02;IN      ;19970101;19990101
       1     2       3              4        5      6       7        8
```

| Note | Description |
|---|---|
| 1 | Specifies the alias name of the currency and is used to locate the entry in the file. |
| 2 | Specifies the International Standards Organization (ISO) currency name. |
| 3 | Provides a description of the current entry. |
| 4 | Indicates the exchange rate to be used. The decimal point can be expressed as a comma or a period. |
| 5 | Specifies the default number of decimal places in the currency. |
| 6 | Indicates the EU status. The valid values are IN, OUT, and OTHER. |
| 7 | Specifies the beginning date for use of this table entry. |
| 8 | Specifies the ending date for use of this table entry. |

**Note:**

If you use a table format that is not similar to the EuroConverter Studio currency table format, you must change the PARSE command in VIAGECTB to accommodate the changed format.

# Step 2 - Invoking Bridge

## Step 2a - Invoking Bridge from an ISPF Panel

To add a Bridge option to the ISPF Primary Menu or another dialog menu, follow this step:

- Use these panel definition statements:

  Use this line to describe the Bridge option to the user:

  ```
  % S +Bridge - Viasoft year 2000 Bridging technology
  ```

  Use this line to invoke Bridge based on the user selection of the letter S:

  ```
  S, 'CMD(%BRIDGE) NEWAPPL(VIAG)'
  ```

**Note:**

After you update the ISPF environment, you may need to re-enter ISPF to make the facilities available.

## Step 2b - Invoking Bridge with a CLIST

To invoke Bridge with a CLIST, follow this step:

- Use the VIABRDG CLIST rather than the BRIDGE CLIST. The VIABRDG CLIST invokes the BRIDGE CLIST while specifying the NEWAPPL(VIAG) parameter to set the correct application ID.

# Step 3 - Validating Bridge

Follow these steps to verify that you have correctly installed Bridge.

## Step 3a - Preparing for Batch Verification

The CNTL library contains installation verification program (IVP) jobs that were tailored during installation, but may still need further modification. Please run them exactly as described below.

To allocate files for the sample programs, follow this step:

- Run the job VIAGAIVB in the CNTL library to allocate files for the sample programs before running the verification procedures.

## Step 3b - Verifying Conversion of Sequential Files

### *To verify conversion of sequential files*

1. Verify the JOB card information in VIAGICBJ.

2. Edit the VIASOFT and CENTER qualifiers, if Center is already installed.

3. Submit VIAGICBJ for forward bridging and verify these results:

   - The job must end with completion code 0.
   - FILE01 should contain all records of type 01. The records should contain three non-converted dates (dates of length 6) in positions 11, 19, and 27.
   - FILE02 should contain all records of type 02. The record should contain three non-converted dates (dates of length 6) in positions 6, 14, and 22.
   - FILE03 should contain all records of type 03. The records should contain two non-converted date (dates of length 6) in positions 11 and 49.

### *To perform reverse bridging*

1. Repeat step 1 and step 2 from To verify conversion of sequential files for VIAGIRBJ.

2. Submit the VIAGIRBJ job for reverse bridging and verify these results:

   - The job must end with completion code 0.
   - FILE01 should contain all records of type 01. The records should contain three non-converted dates (dates of length 8) in positions 11, 21, and 31.
   - FILE02 should contain all records of type 02. The record should contain three non-converted dates (dates of length 8) in positions 6, 16, and 26.
   - FILE03 should contain all records of type 03. The records should contain two non-converted date (dates of length 8) in positions 11 and 51.

## Step 3c - Verifying Conversion of Key Sequenced (KSDS) VSAM Files Processed through Sequential Access Mode

To verify the conversion of key sequenced (KSDS) VSAM files processed through sequential access mode, follow this step:

▶ Submit the VIAGIVSJ job from the CNTL library and verify these results:

- The job must end with completion code 0.
- The print step of this job prints the contents of FILE01 (a VSAM KSDS File). The records should contain three unconverted dates in positions 7, 14, and 75.

## Step 3d - Verifying Conversion of Key Sequenced (KSDS) VSAM Files Processed through Random Access Mode

To verify the conversion of key sequenced (KSDS) VSAM files processed through random access mode, follow this step:

- Submit the VIAGIVRJ job from the CNTL library and verify these results:
  - The job must end with completion code 0.
  - The print step of this job prints the contents of FILE01 (a VSAM KSDS File). The records should contain three unconverted dates in positions 7, 14, and 75.

## Step 3e - Verifying Conversion of Currency Table - Bridge for Euro Option Only

### For Euro Batch

***To compile and link the IVP source program.***

1. Submit the VIAGEBIB CNTL library member to compile and link the IVP source program.
2. Submit VIAGEBIJ.
3. Verify these results:
   - The job must end with completion code 0.
   - The balance for Acct Number AAAA-1111 should be 1.66.
   - The total records processed from this job should be 6.

### For Euro VSAM

***To verify currency table conversion for euro VSAM***

1. Run VIAGEBVD to allocate the IVP file.
2. Call VIAGEBVB to compile and link the IVP source.
3. Submit VIAGEBVJ.
4. Verify these results:
   - The job must end with completion code 0.
   - The balance for Acct Number AAAA-1111 should be 1.66.
   - The total records processed from this job should be 6.

# Installing Bridge in a CICS Environment

You must perform these steps for each CICS environment where you want Bridge to be active. It is not necessary to perform other installation steps as long as all the CICS environments reside at the same host environment.

## Step 1 - Installing CICS Definitions

This step installs the definitions of programs, transactions, etc., necessary to Bridge.

### Step 1a - Defining CICS Option Definitions

#### Running VIAGCSDI

***To define CICS option definitions***

1. Edit member VIAGCSDI.
2. Verify the JOB statement, CICS loadlib, CSD file, and the IVP file are correct and submit the job. The job should end with maximum completion code 4.
3. Verify that all other DEFINE commands completed successfully.

#### Editing VIAGCRCS (or VIAGECSD for Euro Option)

***To add Bridge CICS Record Level API (RLAPI) resource definitions to your CSD***

1. Run VIAGCCVB to compile and link IVP source.
2. Verify that the JOB statement, CICS loadlib, CSD file, and the IVP file are correct in VIAGCRCS.
3. Submit the VIAGCRCS member from the CNTL library.

***To add Bridge for Euro CICS Record Level API (RLAPI) resource definitions to your CSD***

1. Run VIAGECVB to compile and link IVP source.
2. Verify that the JOB statement, CICS loadlib, CSD file, and the IVP file are correct in VIAGECSD.
3. Submit the VIAGECSD member from the CNTL library.

### Editing VIAGCSDD

**Note:**

Skip this step if you are not installing the CICS DL/I Option.

#### *To add the Bridge CICS DL/I resource definitions to your CSD*

1. Edit member VIAGCSDD.

2. Verify that the JOB statement, CICS load library dataset name, and CSD dataset name are correct and submit the job. The job should end with a completion code of 0.

If you are bridging fields with VSAM keys in remote files, the program VIAGCRFI is executed in the remote file-owning region. You must include a CICS definition in the remote file-owning region and the LOADLIB in which VIAGCRFI resides must be in the DFHRPL for that region. No transaction definitions are necessary.

### *Step 1b - Installing Bridge Group*

Add GROUP(VIAGCICS) to your GRPLIST.

**Note:**

If you do not add the GROUP(VIAGCICS) to your GRPLIST, use CEDA to install this group after restarting CICS.

After you complete the previous CSD updates, use RDO with this command to install the Bridge group:

```
CEDA INS GR(VIAGCICS)
```

If you are installing DL/I, use this command:

```
CEDA INS GR(VIAGDLI)
```

These definitions remain valid until the next CICS cold start. To install the definitions regularly, add the Bridge group to the list installed at CICS cold start (declared in the GRPLIST parameter of the SIT).

## *Step 2 - Activating Bridge at CICS Startup*

View Bridge as extended services to those supplied ordinarily by CICS. To serve the programs/transactions which need them, CICS makes these services available upon startup.

You can use one of these methods to activate Bridge at CICS startup. This guide lists them in order of preference (i.e., Method 1 is preferred).

**Note:**

You can skip this step and use Method 3 to manually activate the product until the complete installation is verified.

## Method 1

Add one or both of the PLT entries contained in the VIAGPLTI member of the CNTL library to your Post Initialization Program List Table. Use your standard procedures to assemble and link-edit the table. Note that the second entry is for the CICS DL/I Support option. If installing this option, you must uncomment the entry, or it is ignored when you assemble the PLT.

```
DFHPLT TYPE=ENTRY,PROGRAM=VIAGCSTR BRIDGE
DFHPLT TYPE=ENTRY,PROGRAM=VIAGDSTR BRIDGE DL/I option
```

ASG recommends you use this method of initializing Bridge in your CICS system. If problems occur when you activate Bridge, the system console operator decides whether to proceed with the CICS startup or to shut down. The other methods do not have this property.

## Method 2

Invoke one or both of these transactions via a sequential input device (card reader input):

```
B2KD,ON\   (Initializes VSAM support)
B2KL,ON\   (Initializes Bridge DL/I support)
```

## Method 3

This is the same as Method 2 except that it is invoked manually rather than via a sequential device. After restarting CICS (see "Step 3 - Validating CICS Installation" on page 30.), invoke one or both of these transactions from any terminal:

```
B2KD,ON to initialize the VSAM support.
B2KL,ON to initialize the DL/I support.
```

**Note:**

ASG strongly recommends securing these transactions at a later time. Inadvertent use of them might inactivate Bridge while CICS is still operating.

### Results

If you activate Bridge automatically, you should find this message in the CICS log:

```
ASG3517I BRIDGE INITIALIZED AND SET TO ON.
```

If you activate Bridge manually using the syntax, `B2KD,ON to initialize the VSAM support,` you should receive this message at your terminal:

```
ASG-Bridge: Set to ON successfully completed
```

If you activate Bridge manually using the syntax, `B2KL,ON to initialize the DL/I support,` you should receive this message at your terminal:

```
ASG-Bridge (CICS/DLI): Set to ON successfully completed
```

## Step 3 - Validating CICS Installation

These steps verify that you have correctly installed Bridge in a CICS environment.

### Step 3a - Allocating the Installation Verification File

In this step you allocate a file that is needed for the installation verification stage performed in the next step.

Submit the VIAGAIVC member from the CNTL library after verifying that the JOB statement, IVP file, and VOLSER were correctly changed by the tailoring step of the build job run previously. The IVP file name must be the same as the one defined earlier in "Step 1 - Modifying and Executing CNTL Library Members" on page 11 .

### Step 3b - Allocating and Populating the IVP Libraries

**Note:**

If you are not installing the CICS DL/I Support option, skip steps Step 3b - Allocating and Populating the IVP Libraries through "Step 3e - Starting the Trace Facility" on page 31.

***To allocate several small libraries required for the IVP process***

**1** Edit the VIAGICPD member from the CNTL library, verify that the JOB statement is correct, and modify the JCL and parameters as necessary to suit your installation requirements.

**2** Save and submit the job. The job should end with a completion code of 0.

## Step 3c - Allocating and Initializing the Sample IVP Database

***To allocate and initialize a sample IVP database that is used by the forward and reverse bridging CICS DL/I IVP tests***

1. Edit members VIAGIDFJ and VIAGIDRJ from the CNTL library, verify that the JOB statement is correct, and modify the JCL and parameters as necessary to suit your installation requirements.

2. Save and submit each job. Both jobs should end with a completion code of 0.

## Step 3d - Generating Bridge ACBs

Member VIAGACBJ of the CNTL library generates two additional ACBs required for the IVP.

To edit VIAGACBJ, follow this step:

- Verify the JOBS statement and library dataset names, and submit the job. The job should end with a completion code of 0.

## Step 3e - Starting the Trace Facility

The Bridge CICS DL/I Support feature provides a special trace facility. This facility issues trace records, similar to those seen on a VIAGTRAC log, pertaining to DL/I bridging to the message log.

Bridge provides a transaction to control the scope and content of the trace. This is the syntax of the transaction:

```
B2KT {ON|OFF} [,TERM=terminal-id] [,TRACE=pgm-name[,xx]]
```

where:

| Parameter Value | Description |
|---|---|
| ON | Starts the trace facility. |
| OFF | Stops the trace facility. |
| `terminal-id` | Specifies a CICS terminal for tracing. This parameter traces Bridge activities only for the specified terminal. To trace all terminals, specify TERM=****. |
| `pgm-name` | Specifies the Bridge programs traced. The `xx` subparameter specifies the type of trace message you want. See the TRACE= parameter in the Controlling Bridge Trace Log Content section of the *ASG-Bridge User's Guide.* |

## Step 4 - Editing the CICS Startup JCL

In this step you add the Bridge load library, the Bridge definition generated modules library, and the DCR and PCR libraries (if the CICS DL/I option is installed) to the CICS startup JCL.

Modify these JCL statements and add them to the CICS startup JCL:

```
//DFHRPL DD DISP=SHR,DSN=CICS application libraries
//    DD DISP=SHR,DSN=Bridge product load libraries
//    DD DISP=SHR,DSN=Bridge generated libraries
//    DD DISP=SHR,DSN=DCR library
//    DD DISP=SHR,DSN=PCR library
```

**Note:**

Use the last two statements only if you are installing the CICS DL/I option.

## Step 5 - Adding Bridge Entries to your CICS DL/I Database Directory (DDIR)

**Note:**

You can skip this step if you are not using Local DL/I and your site does not require or use a DL/I Database Directory (DDIR).

Member VIAGLDBD of the CNTL library contains additional DDIR entries for the IVP database.

To insert these entries into your DDIR table, follow this step:

- Copy the contents of VIAGLDBD into your DDIR and regenerate it.

## Step 6 - Adding Bridge Entries to your CICS DL/I PSB Directory (PDIR)

**Note:**

If you are not installing the CICS DL/I Support option, skip this step.

Member VIAGLPSB of the CNTL library contains additional PDIR entries for the IVP database.

To insert these entries into your PDIR table, follow this step:

- Copy the contents of VIAGLPSB into your PDIR, then regenerate it.

## Step 7 - Restarting CICS

Restart CICS to activate the changes you have just installed. If you prefer to manually activate Bridge, as explained in "Method 3" on page 29, activate it now. Bridge is ready for use.

If you activate Bridge automatically, you should find this message in the CICS log:

```
ASG3517I BRIDGE INITIALIZED AND SET TO ON.
```

If you activate it manually, you should receive this message at your terminal:

```
Bridge: Set to ON successfully completed
```

## Step 8 - Verifying CICS Installation

### *To verify CICS installation*

1. Verify that all components of Bridge are properly installed and enabled in the CICS environment.
2. Invoke the transaction B2KI. The transaction tests a simple scenario and reports on each step of the test. A successful verification ends with this message:

   ```
   Installation verification of Bridge successfully
   completed.
   ```

3. Invoke the transaction B2KR. The transaction tests a simple scenario and reports on each step of the test. A successful verification ends with this message:

   ```
   Installation verification of Bridge successfully
   completed.
   ```

The screen also displays any errors. You can find additional details in the CICS log. Consult your organization's technical support regarding any problems.

## Step 9 - Running the IVP Transactions

**Note:**

If you are not installing the CICS DL/I Support option, skip this step.

This step executes two IVP transactions. Each transaction verifies that the CICS DL/I Support option is properly installed and enabled in the CICS environment.

### *To run the IVP transactions*

**1** Invoke the B2KC transaction to test the forward bridging functions of the option. The transaction displays a number of messages as it executes, and should end with this message:

```
Installation verification of Bridge/DLI successfully
completed
```

**2** Invoke the B2KV transaction to test the reverse bridging functions of the option. The transaction displays a number of messages as it executes, and should end with this message:

```
Installation verification of Bridge/DLI successfully
completed
```

This completes the CICS DL/I Support option installation verification.

## 9a. Running IVP Transaction for CICS RLAP

To test the CICS RLAPI functions of the option, follow this step:

- Invoke the BRIV transaction. The transaction displays this message when it has executed correctly:

```
RECORD LEVEL API IVP
NPUT RECORD
960323
OUTPUT RECORD
19960323
PROCESSING COMPLETED
```

## 9b. Running the IVP Transaction for Euro Option

To test the Bridge for Euro functions of the option, follow this step:

- Invoke the BEUR transaction. The transaction displays this message when it has executed correctly:

```
PROGRAM STARTED
ASG-BRIDGE INITIALIZED.
FILES OPENED.
FOLLOWING ARE DISPLAYS OF THE INPUT FILE.

ACCT NUMBER CHECK BAL CHECK RL SAVE BAL SAVE RL OVER BAL OVER RL OVER MAX
AAAA-1111      1.67   .00      52.72   .00     43.75   .02     76.22
AAAA-2222    152.61   .01      33.97   .00     74.24   .02     76.22
AAAA-3333    305.20   .02      67.82   .03     51.38   .03     76.22
AAAA-4444    507.66   .02      84.74   .01     34.61   .03     76.22
```

```
BBBB-1111    677.02   .01      101.66   .01     28.51   .01     76.22
BBBB-2222    846.10   .00      118.59   .03     13.26   .02     76.22

TOTAL RECORDS PROCESSED :   00006
PROGRAM ENDED.
```

# Validating Bridge in an IMS Environment

**Note:**

These steps only apply if you are validating the Bridge IMS Support option.

This process allocates several small libraries and a sample IMS database and then runs an IVP program to validate that the Bridge IMS Support option is installed correctly.

## *Step 1 - Allocating and Populating the IMS IVP Libraries*

### *To allocate several small libraries required for the IMS IVP process*

**1** Edit the VIAGICPY member from the CNTL library and verify that the JOB statement is correct.

**2** Save and submit the job. The job should end with a completion code of 0.

## *Step 2 - Allocating and Initialize the IMS IVP Database*

### *To allocate and initialize the IMS databases required for the IMS IVP process*

**1** Edit the VIAGILIJ member from the CNTL library, verifying that the JOB statement is correct.

**2** Save and submit the job. The job should end with a completion code of 0.

## *Step 3 - Running the IMS IVP Program*

This step runs the IMS IVP program. It processes the sample database invoking Bridge to convert fields contained in the database to an expanded format.

### *To run the IMS IVP program*

**1** Edit the VIAGIRIJ member from the CNTL library, verifying that the JOB statement is correct.

**2** Save and submit the job. The job should end with a completion code of 0.

## Step 4 - Running the IMS/HSSR Program

**Note:**
Complete this step if your site plans to use Bridge with IMS/HSSR programs.

This step runs the IMS/HSSR IVP program. It processes the sample database created in "Step 2 - Invoking Bridge" on page 24, invoking Bridge to convert fields contained in the database to an expanded format.

### *To run the IMS/HSSR IVP program*

**1** Edit the VIAGIHRJ member from the CNTL library, verifying that the JOB statement is correct.

**2** Save and submit the job. The job should end with a completion code of 0.

# Validating Bridge in an IMS FastPath Environment

Bridge supports DEDB (Data Entry DataBase) databases in both Batch and online environments. This support is equivalent to that provided for standard full function DL/I databases. In addition, Bridge supports MSDB (Main Storage DataBase) databases in both Fast Path Potential (MPP) and Fast Path Exclusive (IFP) environments.

The supplied IVP program uses an MSDB database as input to execute as a Fast Path Exclusive transaction.

## Step 1 - Defining the Transaction, Database, and PSB

This step must be performed by the installation IMS systems programmer. Add these definitions to the source for IMS stage1 and run a stage1 and stage2 IMS generation to define the IVP transaction and database to the target IMS system. Member VIAGXTRN in the CNTL library also contains these definitions:

```
*******************************************************
*   ASG-BRIDGE FASTPATH IVP TRANSACTION AND MSDB
    DATABASE DEFINITION
*******************************************************
*
         DATABASE DBD=DPODYF04,ACCESS=UP
*
         APPLCTN RESIDENT,PSB=PODYOF06,FPATH=256
         TRANSACT CODE=ODYF6,MODE=SNGL,                                 *
               MSGTYPE=(SNGLSEG,RESPONSE,1)
*
```

## Step 2 - Performing a DBDGEN, PSBGEN, and ACBGEN for Supplied DBD and PSB

The installation IMS Systems Programmer usually performs this step. CNTL library members VIAGXFPD and VIAGXFPP contain the source for the DBD and PSB you use with the FastPath IVP transaction.

### *To perform a DBDGEN, PSBGEN, and ACBGEN for supplied DBD and PSB*

1. Copy into the installation IMS libraries containing this type data.
2. Execute a DBDGEN and PSBGEN for the DBD and PSB, respectively.
3. Execute an ACBGEN to place the ACB into the staging ACBLIB for the target IMS system.
4. Use the existing installation procedure to copy the ACB into the active ACBLIB of the target IMS system.

## Step 3 - Performing an MFSGEN to Install Supplied MFS Format

This step will normally be performed by the installation IMS systems programmer. CNTL library member VIAGXMFS contains the source of the MFS format to be used with the FastPath IVP.

### *To perform an MFSGEN to install supplied MFS format*

1. Copy VIAGXMFS into the appropriate installation MFS source library.
2. Execute an MFSGEN for this format into the target IMS system staging format library.
3. Use the existing installation procedure to copy the format into the active format library.

## Step 4 - Creating the MSDB Database Initialization File

This step is normally performed by the installation IMS systems programmer. CNTL library member VIAGXDBJ contains sample JCL for creating an initialization file for the IVP MSDB. If the installation has existing MSDB databases, refer to the comments in member VIAGXDBJ before proceeding with this step. For further information refer to IMS documentation for program DBFDBMA0.

To create the MSBD database initialization file, follow this step:

- Add this statement to the installation IMS.PROCLIB member DBFMSDB*x* (where *x* is the suffix of the member currently used by the target IMS system):

```
DBD=DPODYF04,NBRSEGS=3,F
```

## Step 5 - Creating the JCL for Executing the IFP Region for an IVP Transaction Program

This step is normally performed by the installation IMS systems programmer. CNTL library member VIAGXFPJ contains sample JCL for executing an IMS FastPath region.

This JCL should only be used as an example because many of the jobstep parameters are installation dependent.

To create the JCL for executing the IFP region for an IVP transaction, follow this step:

- Copy the load module VIAGIFP1 from the Bridge LOADLIB to the installation load library used for IMS programs, or specify the Bridge LOADLIB as the STEPLIB in the IFP region JCL.

## Step 6 - Performing Bridge DCRGEN and PCRGEN for Supplied DBD and PSB

### *To perform Bridge DCRGEN and PCRGEN for supplied DBD and PSD*

1. Modify the symbolic parameters in the cataloged procedure VIAGJGLD. Run a DCRGEN by submitting VIAGJGDC from the CNTL library.

2. Run a DBDGEN and PSBGEN referencing CNTL library members VIAGXFND and VIAGXFNP, respectively, targeting the output to the Bridge new DBD and PSB libraries.

3. Run a PCRGEN by using the PSB you created in the installation PSBLIB earlier as the old PSB and the PSB you created in step 2 as the new PSB.

## Step 7 - Initializing IMS Control Region and Invoking IVP Transaction

The IVP demonstrates that while an IMS database containing two-digit year dates is used, dates displayed to, and accepted from, the IMS terminal operator are properly bridged to an expanded format.

### *To initialize the IMS control region and invoke the IVP transaction*

1. Initialize the IMS control region to load the MSDB database into memory and incorporate the transaction definition into the active IMS system.

2. From an IMS terminal, type `/FOR MODYF6` and press Enter.

3. Type `1` in the PROCESS CODE field, `1234` in the SALE-NO field, and press Enter.

   Sale data and the `SALE FOUND` message displays.

4. Type `2` in the PROCESS CODE field, modify the value in STOCDATE to a valid YYYYDDD date, and press Enter. The message `STOCKDATE UPDATED` displays.

5. Type `3` in the PROCESS CODE field, `1234` in the SALE-NO field, and press Enter. The message `NEXT SALE FOUND` displays with 3473 in the SALE-NO field.

# Installing Bridge in an IDMS Environment

**Note:**

This step is only applicable when installing the Bridge IDMS Support option.

## *Step 1 - Adding Attributes and Protocols Required by Bridge to the IDMS Data Dictionary*

CNTL member VIAGMPRB contains these Bridge IDMS attributes and protocols for IDMS-DB and IDMS-DC processing with Bridge:

| Bridge Protocol Name | Corresponding CA Protocol Name | Remarks |
|---|---|---|
| VIAGIDB | BATCH | For Batch programs without automatic verification of the ERROR-STATUS field. |
| VIAGIDB-AUTO | BATCH-AUTOSTATUS | For Batch programs with automatic verification of the ERROR-STATUS field. |

| Bridge Protocol Name | Corresponding CA Protocol Name | Remarks |
|---|---|---|
| VIAGIDC | IDMS-DC-NONAUTO | For online programs without automatic verification of the ERROR-STATUS field. |
| VIAGIDC-AUTO | IDMS-DC | For online programs with automatic verification of the ERROR-STATUS field. |

Have your IDMS DBA complete these steps for each dictionary that will be used to precompile IDMS programs for use with Bridge:

- Add these attributes and protocols
- Assign the attributes VIAGIDC and VIAGIDC-AUTO to the record SUBSCHEMA-CTRL that is currently connected to the attributes IDMS-DC and IDMS-DC-NOAUTO

## Step 2 - Relinking Bridge with the IDMS Interface

***To relink the Bridge IDMS interface program to include a local copy of the IDMS interface module***

1. Edit member VIAGLIDM from the CNTL library and verify that the JOB statement is correct. If necessary, modify any parameters.
2. Save and submit the job. The job should end with a completion code of 0.

## Step 3 - Validating IDMS Support Option Installation

With this step you allocate a sample IDMS database and run an IVP Batch program and an IVP online transaction to validate that the Bridge IDMS Support option installed correctly.

**Note:**

To ensure proper installation, the IDMS database administrator should perform steps "Step 3a - Preparing for Batch Verification" on page 24, through "Step 3c - Verifying Conversion of Key Sequenced (KSDS) VSAM Files Processed through Sequential Access Mode" on page 25.

## Step 3a - Allocating the IDMS IVP Database

This step allocates a small IDMS database required for the IDMS IVP process. This step is required for the IDMS-DB and IDMS-DC options of Bridge.

### *To allocate the IDMS IVP database*

1. Edit the VIAGMCPY member from the CNTL library and verify that the JOB statement is correct. If necessary, modify any parameters.

2. Save and submit the job. The job should end with a completion code of 0.

## Step 3b - Defining IDMS Database to the IDMS Dictionary

### *To define the IDMS database to the IDMS dictionary*

1. Use these CNTL members, keeping this information in mind:
   - You are defining a database with short dates (6 positions)
   - The description in the programs reflects longer dates (8 positions)
   - The programs access the short dates

| CNTL Name | Description | Remarks |
|---|---|---|
| VIAGMOSG | Describes, in a segment, the physical file of the database. | Eventually change the page range |
| VIAGMDMC | Changes to the IDMS global DMCL. | No new segment needed |
| VIAGMDBT | Defines the database name table for the IDMS database. | |
| VIAGMELM | Defines all the data elements for the databases. | |
| VIAGMORC | Defines records with version 1 describing the physical database. | |
| VIAGMOSC | Defines the schema IDMSCM with version 1 for the physical database. | |
| VIAGMOSB | Defines the subschema IDMTOBAL for use at run time. | |

| CNTL Name | Description | Remarks |
|---|---|---|
| VIAGMNRC | Defines records with version 5 describing the program's view of the database. | |
| VIAGMNSC | Defines the schema IDMSCM with version 5 for the program's view of the database. | |
| VIAGMNSB | Defines the subschema IDMTNBAL for the program's view of the database. | |
| VIAGSGEN | Defines the programs and task for IDMS-DC option of Bridge. | |

**2** Add the necessary ESW load libraries (including SCR libraries and DCR libraries) to the CDMSLIB DD statement in the startup JCL of the IDMS Central Version.

**3** Add the DD statement for the IVP database file that was allocated in "Step 3a - Preparing for Batch Verification" on page 24, to the startup JCL of the IDMS Central version.

**4** Shutdown the Central Version.

**5** Restart the Central Version.

## Step 3c - Initializing and Loading the IDMS IVP Database

### *To initialize and load the IDMS database required for the IDMS IVP process*

**Note:**

This step is required for the IDMS-DB and IDMS-DC options of Bridge.

**1** Edit the VIAGMLIJ member from the CNTL library and verify that the JOB statement is correct. If necessary, modify any member parameters.

**2** Save and submit the job. The job should end with a completion code of 0 and perform these tasks:

- Vary the IVP database offline in the Central Version
- Format (initialize) the IVP database
- Vary the IVP database online in the Central Version
- Execute the load program VIAGMIVL

This is a sample of the output expected from the load program.

```
STORED MST-TEACHER =======> 00001A980301
STORED MST-CLASS-TEACHR ==> 001
STORED MST-STUDENT =======> MALDER    001
STORED MST-SUBJECT =======> 00001
STORED MST-GRADES ========> 0003
STORED MST-SUBJECT =======> 00002
STORED MST-GRADES ========> 0003
STORED MST-SUBJECT =======> 00003
STORED MST-GRADES ========> 0003
STORED MST-SUBJECT =======> 00004
STORED MST-GRADES ========> 0003
STORED MST-SUBJECT =======> 00005
STORED MST-GRADES ========> 0003
STORED MST-STUDENT =======> SKALI     003
STORED MST-SUBJECT =======> 00001
STORED MST-GRADES ========> 0003
STORED MST-SUBJECT =======> 00002
STORED MST-GRADES ========> 0003
STORED MST-SUBJECT =======> 00003
STORED MST-GRADES ========> 0003
STORED MST-SUBJECT =======> 00004
STORED MST-GRADES ========> 0003
STORED MST-SUBJECT =======> 00005
STORED MST-GRADES ========> 0003
STORED MST-STUDENT =======> MADONA    002
STORED MST-SUBJECT =======> 00001
STORED MST-GRADES ========> 0003
STORED MST-SUBJECT =======> 00002
STORED MST-GRADES ========> 0003
STORED MST-SUBJECT =======> 00003
STORED MST-GRADES ========> 0003
STORED MST-SUBJECT =======> 00004
STORED MST-GRADES ========> 0003
STORED MST-SUBJECT =======> 00005
STORED MST-GRADES ========> 0003
```

## Step 3d - Running the IDMS IVP Program

This step processes the sample IDMS database invoking Bridge IDMS to convert short fields contained in the database to an expanded format.

### *To run the IDMS IVP program for the IDMS-DB option of Bridge*

1. Edit the VIAGMRIJ member from the CNTL library and verify that the JOB statement is correct. If necessary, modify any member parameters.

2. Save and submit the job. The job should end with a completion code of 0 and perform these tasks:

   - Make a report of the TEACHER records in the IVP database
   - Manipulate various other records in the IVP database
   - Rollback the updates

This is a sample of the output expected from the IVP program:

```
++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++ AUDIT OF TEACHER +++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++
+ TEACHER = 00048MELISA    B19980301
+ CALC    = 00048MELISA    B19980301
+ TEACHER = 00041BILL      B20281228
+ CALC    = 00041BILL      B20281228
+ TEACHER = 00001ROGER     A19980301
+ CALC    = 00001ROGER     A19980301
+ TEACHER = 00022HENRY     X20270427
+ CALC    = 00022HENRY     X20270427
+ TEACHER = 00002PETER     A20011213
+ CALC    = 00002PETER     A20011213
+ TEACHER = 00055JON       A20440601
+ CALC    = 00055JON       A20440601
+ TEACHER = 00024KELY      A20030903
+ CALC    = 00024KELY      A20030903
+ TEACHER = 00023ANNY      X20250922
+ CALC    = 00023ANNY      X20250922
+ TEACHER = 00003DONALD    X20000411
+ CALC    = 00003DONALD    X20000411
+ TEACHER = 00008JUDITH    X20111201
+ CALC    = 00008JUDITH    X20111201
+ TEACHER = 00011SHERON    B20460222
+ CALC    = 00011SHERON    B20460222
+ TEACHER = 00019BRENDON   X19550611
+ CALC    = 00019BRENDON   X19550611
+ TEACHER = 00033MIKEL     A20221005
+ CALC    = 00033MIKEL     A20221005
+ TEACHER = 00006MIKI      X20000321
+ CALC    = 00006MIKI      X20000321
+ TEACHER = 00005KREMER    A19991231
+ CALC    = 00005KREMER    A19991231
++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++ AUDIT OF TEACHER +++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++
          F O R W A R D   B R I D G I N G
+++++++++++++++++++++++++++++++++++++++++++++++++
+ TEACHER +++++++++++++++++++++++++++++++++++++++
READING  MST-TEACHER, CALC KEY = 00001A19980301
RESULT   MST-TEACHER = 00001ROGER     A19980301
MODIFIED MST-TEACHER = 00001ROGER     A19991231
READING  MST-TEACHER, CALC KEY = 00001A19991231
RESULT   MST-TEACHER = 00001ROGER     A19991231
+ CLASS +++++++++++++++++++++++++++++++++++++++++
MODIFIED MST-CLASS-TEACHR = 001DAVID     001002003004005006007008009010011012O
READING  MST-CLASS-TEACHR = 005ROGER     501502503504005006007008009010011012O
MODIFIED MST-CLASS-TEACHR = 005DAVID     501502503504005006007008009010011012O
MODIFIED MST-CLASS-TEACHR = 002DAVID     201202203504005006007008009010011012O
+ STUDENT +++++++++++++++++++++++++++++++++++++++
STORED   MST-STUDENT = 999CORNETTE
+ SUBJECT +++++++++++++++++++++++++++++++++++++++
READING  MST-SUBJECT WITH INDEX = 19850101
READING  MST-SUBJECT = 0000400319850101088
MST-DATE-LAST-GRADE IN MST-SUBJECT = 19850101
READING  MST-SUBJECT = 0000400319850101088
READING  MST-SUBJECT = 0000400319850101088
READING  MST-SUBJECT = 0000400319850101088
+ TEACHER +++++++++++++++++++++++++++++++++++++++
READ IX1 MST-TEACHER = 00019BRENDON   X19550611
READ IX  MST-TEACHER = 00001ROGER     A19991231
+ STUDENT +++++++++++++++++++++++++++++++++++++++
READ IX  MST-STUDENT = 103SKALI
+ SUBJECT +++++++++++++++++++++++++++++++++++++++
READING LAST MST-SUBJECT = 0000200319980123060
+ GRADE +++++++++++++++++++++++++++++++++++++++++
READING FIRST MST-GRADES =   20101212045198807070302044050508120 20
MODIFIED MST-GRADES =  201012120451988070703020000101030196701011002 0
++++++++++++++++++++++++++++++++++++++++++++++++++
ROLLBACK STARTING NOW ...
ROLLBACK WAS OK ...
```

## Step 3e - Compiling and Running the IDMS-DC IVP Program

This step processes the sample IDMS database invoking Bridge to convert short fields contained in the database to an expanded format.

To run the IVP program for the IDMS-DC option, follow this step:

- Connect to the IDMS Central Version and execute task VIAGIVP.

  This is a sample output expected from this task:

```
+ PROGRAM STARTING ++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++ AUDIT OF TEACHER ++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++
+ TEACHER =                        00048MELISA    B19980301
+ CALC    =                        00048MELISA    B19980301
+ TEACHER =                        00041BILL      B20281228
+ CALC    =                        00041BILL      B20281228
+ TEACHER =                        00001ROGER     A19980301
+ CALC    =                        00001ROGER     A19980301
+ TEACHER =                        00022HENRY     X20270427
+ CALC    =                        00022HENRY     X20270427
+ TEACHER =                        00002PETER     A20011213
+ CALC    =                        00002PETER     A20011213
+ TEACHER =                        00055JON       A20440601
+ CALC    =                        00055JON       A20440601
+ TEACHER =                        00024KELY      A20030903
+ CALC    =                        00024KELY      A20030903
+ TEACHER =                        00023ANNY      X20250922
+ CALC    =                        00023ANNY      X20250922
+ TEACHER =                        00003DONALD    X20000411
+ CALC    =                        00003DONALD    X20000411
+ TEACHER =                        00008JUDITH    X20111201

+ CALC    =                      00008JUDITH    X20111201
+ TEACHER =                      00011SHERON    B20460222
+ CALC    =                      00011SHERON    B20460222
+ TEACHER =                      00019BRENDON   X19550611
+ CALC    =                      00019BRENDON   X19550611
+ TEACHER =                      00033MIKEL     A20221005
+ CALC    =                      00033MIKEL     A20221005
+ TEACHER =                      00006MIKI      X20000321
+ CALC    =                      00006MIKI      X20000321
+ TEACHER =                      00005KREMER    A19991231
+ CALC    =                      00005KREMER    A19991231
+++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++ AUDIT OF TEACHER +++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++
          F O R W A R D   B R I D G I N G
++++++++++++++++++++++++++++++++++++++++++++++++++
+ TEACHER ++++++++++++++++++++++++++++++++++++++++
READING  MST-TEACHER, CALC KEY =   00001A19980301
RESULT   MST-TEACHER =            00001ROGER     A19980301
MODIFIED MST-TEACHER =            00001ROGER     A19991231
READING  MST-TEACHER, CALC KEY =   00001A19991231
RESULT   MST-TEACHER =            00001ROGER     A19991231
+ CLASS ++++++++++++++++++++++++++++++++++++++++++
         READING  MST-CLASS-TEACHR =      005
MODIFIED MST-CLASS-TEACHR =        005
+ STUDENT ++++++++++++++++++++++++++++++++++++++++
STORED   MST-STUDENT =            999CORNETTE
+ SUBJECT ++++++++++++++++++++++++++++++++++++++++
READING  MST-SUBJECT WITH INDEX =  19850101
READING  MST-SUBJECT =            0000400319850101088
 MST-DATE-LAST-GRADE IN MST-SUBJECT19850101
READING  MST-SUBJECT =            0000400319850101088
READING  MST-SUBJECT =            0000400319850101088
READING  MST-SUBJECT =            0000400320111111088
+ TEACHER ++++++++++++++++++++++++++++++++++++++++
READ IX1 MST-TEACHER =            00019BRENDON   X19550611
READ IX  MST-TEACHER =            00001ROGER     A19991231
+ STUDENT ++++++++++++++++++++++++++++++++++++++++
READ IX  MST-STUDENT =            103SKALI
+ SUBJECT ++++++++++++++++++++++++++++++++++++++++
READING LAST MST-SUBJECT =         0000200319980123060
+ GRADE ++++++++++++++++++++++++++++++++++++++++++
READING FIRST MST-GRADES =         20101212045
MODIFIED MST-GRADES =             20101212045
++++++++++++++++++++++++++++++++++++++++++++++++++
ROLLBACK STARTING NOW ...
          ROLLBACK WAS OK ...
+ PROGRAM ENDING +++++++++++++++++++++++++++++
V12  ENTER NEXT TASK CODE:
```

# Appendix A

# Bridge CNTL and CLIST Library Members

## Bridge CNTL Library Members

These are the Bridge CNTL library members:

| Member | Description |
|---|---|
| VIA$PRMG | Bridge installation parameters |
| VIA$PRMU | Bridge installation parameters |
| VIAGABLJ | JCL to allocate and initialize the Bridge Event Log |
| VIAGACBJ | ACB Generate for the IMS IVP database |
| VIAGAIVB | JCL to allocate the installation verification Batch file |
| VIAGAIVC | JCL to allocate the installation verification CICS file |
| VIAGAKRJ | JCL to allocate the Bridge Management AKR |
| VIAGASMP | JCL to assemble the Bridge run-time parameters module |
| VIAGBPSS | Bridge parameters starter set |
| VIAGBRSS | Bridge Rules starter set |
| VIAGBRGP | Bridge install macro |
| VIAGCAKR | JCL to export an AKR from a prior release of Bridge |
| VIAGCCVB | Job to compile and link COBOL IVP source for CICS RLAPI |
| VIAGCDCB | Installation verification COBOL source |
| VIAGCDRB | Installation verification COBOL source |
| VIAGCDVR | Installation verification COBOL source |
| VIAGCDVS | Installation verification COBOL source |

| Member | Description |
|---|---|
| VIAGCIVF | Installation verification CICS program |
| VIAGCIVR | Installation verification CICS program |
| VIAGCPRT | Installation verification copybook source |
| VIAGCPR1 | Installation verification copybook source |
| VIAGCPR2 | Installation verification copybook source |
| VIAGCPR3 | Installation verification copybook source |
| VIAGCPSI | Installation verification copybook source |
| VIAGCPSO | Installation verification copybook source |
| VIAGCSDD | Job to install CSD definitions for the CICS DL/I Support option |
| VIAGCSDI | Job to install CSD definitions for VSAM file support |
| VIAGDMG | Installation verification program input data |
| VIAGDMGR | Installation verification program input data |
| VIAGDRNO | Installation verification program input data |
| VIAGDSQO | Installation verification program input data |
| VIAGEBIB | JCL to compile batch sequential COBOL IVP |
| VIAGEBIJ | JCL to run batch sequential COBOL IVP |
| VIAGEBVB | JCL to compile and link batch VSAM COBOL IVP |
| VIAGEBVD | Job to allocate the installation verification VSAM file for Bridge euro |
| VIAGEBVJ | Installation verification VSAM KSDS (Sequential) |
| VIAGECVB | JCL to compile Bridge euro CICS IVP |
| VIAGECVS | Euro CICS IVP program |
| VIAGEPRM | User exit parameter area |
| VIAGFIVJ | Field level API IVP job |
| VIAGFIVP | Field level API IVP program source |
| VIAGFLAA | Field level API IVP parameters - Assembler |
| VIAGFLAC | Field level API IVP parameters - COBOL |

| Member | Description |
|---|---|
| VIAGFLAD | Field level API Bridge Rule table DSECT |
| VIAGFLAG | Field level API Bridge Rule table generate job |
| VIAGFLAM | Field level API Bridge Rule table generate macro |
| VIAGFLAP | Field level API parameters - PL/I |
| VIAGFLAS | Field level API Bridge Rule table entries |
| VIAGICBJ | Installation verification job for COBOL sequential access |
| VIAGICPD | Allocates CICS DL/I IVP libraries |
| VIAGICPY | Allocates IMS IVP libraries |
| VIAGIDFI | Input to VIAGIDFJ |
| VIAGIDFJ | CICS DL/I forward bridging IVP job |
| VIAGIDRI | Input to VIAGIDRJ |
| VIAGIDRJ | CICS DL/I reverse bridging IVP job |
| VIAGIGNR | Bridge Routine macro |
| VIAGILII | Sample data for IMS IVP database |
| VIAGILIJ | Loads IMS IVP database |
| VIAGIRBJ | Installation verification job for COBOL reverse bridge |
| VIAGIRII | Input data for IMS IVP job |
| VIAGIRIJ | IMS IVP job |
| VIAGIVRJ | Installation verification job for VSAM KSDS random access |
| VIAGIVSJ | Installation verification job for VSAM KSDS sequential access |
| VIAGJGBS | Generate sequential/VSAM Bridge Routine JCL |
| VIAGJGBI | Generate IMS Bridge Routine JCL |
| VIAGJGIP | IMS PCR generate JCL |
| VIAGJGLD | IMS logical DBD Bridge Routine generate JCL PROC |
| VIAGLDBD | Bridge DDIR entries |
| VIAGLIDM | Link Bridge IDMS interface |
| VIAGLPAC | Copy Bridge ESR to MLPA/PLPA |

| Member | Description |
|---|---|
| VIAGLPSB | Bridge PDIR entries |
| VIAGMICPY | Allocates and populates IDMS IVP libraries |
| VIAGMLIJ | Allocate and load the IDMS IVP database |
| VIAGMLII | Sample data for the IDMS IVP database |
| VIAGMPRB | Bridge IDMS protocol for Batch processing |
| VIAGMRIJ | Run the IDMS IVP program |
| VIAMRII | Sample data for the IDMS IVP program |
| VIAGMSG | Messages macro |
| VIAGMSGA | Job to assemble the message table |
| VIAGPARM | Bridge run-time parameters |
| VIAGPLTI | PLT for CICS |
| VIAGXFPJ | Sample JCL for executing an IMS FastPath region |
| VIAGSTAB | Message table source |
| VIAGTEST | Example JCL of VIAGTEST execution |
| VIAGUEXI | Sample user exit |

## Bridge CLIST Members

These are the Bridge CLIST members:

| Member | Description |
|---|---|
| BRIDGE | Used by the VIABRDG CLIST to invoke the Bridge product from native TSO. |
| VIABRDG | Invokes the Bridge program from a CLIST by using the correct application ID (VIAG). |
| VIAGBRO | Invokes ISPF Browse used by Bridge Interactive Component. |
| VIAGRXAP | The REXX RLAPI IVP source program. |
| VIAGTEST | Invokes the Bridge program under TSO test for diagnostic purposes only. |

# Index

ASG Worldwide Headquarters Naples Florida USA | asg.com